Multi-agent Reinforcement Learning for Multi-area Power Exchange

Jiachen Xi*, Alfredo Garcia*, Yu Christine Chen†, Roohallah Khatami‡
*Department of Industrial & Systems Engineering, Texas A&M University, College Station, TX, United States
{jx3297, alfredo.garcia}@tamu.edu
†Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada
chen@ece.ubc.ca
‡School of Electrical, Computer, and Biomedical Engineering, Southern Illinois University, Carbondale, IL, United States
roohallah.khatami@siu.edu

***Abstract*—Increasing renewable integration leads to faster and more frequent fluctuations in the power system net-load (load minus non-dispatchable renewable generation) along with greater uncertainty in its forecast. These can exacerbate the computational burden of centralized power system optimization (or market clearing) that accounts for variability and uncertainty in net load. Another layer of complexity pertains to estimating accurate models of spatio-temporal net-load uncertainty. Taken together, decentralized approaches for learning to optimize (or to clear a market) using only local information are compelling to explore. This paper develops a decentralized multi-agent reinforcement learning (MARL) approach that seeks to learn optimal policies for operating interconnected power systems under uncertainty. The proposed method incurs less computational and communication burden compared to a centralized stochastic programming approach and offers improved privacy preservation. Numerical simulations involving a three-area test system yield desirable results, with the resulting average net operation costs being less than 5% away from those obtained in a benchmark centralized model predictive control solution.**

***Index Terms*—Power system, reinforcement learning, uncertainty, decentralized algorithm, actor-critic algorithm**

I. Introduction

Optimizing the total operational costs of interconnected power systems, amidst uncertain conditions, poses a significant challenge for system operators. The uncertainty stems mainly from the unpredictable nature of non-dispatchable renewable energy sources, such as solar and wind power, coupled with potential inaccuracies in load forecasting [1], [2]. This challenge becomes even more daunting in multi-stage problems for which uncertainty may grow over the time horizon of interest. These factors may considerably undermine the power system's reliable and efficient operation, potentially leading to substantial economic loss. A robust optimization approach can be used to identify solutions that guarantee the reliability of the system. However, for multi-stage problems with compounding uncertainty, this usually comes at the expense of solutions that tend to be overly conservative [3]. Stochastic programming techniques such as stochastic dual decomposition (SDDP) have proven quite useful in energy planning problems [4]. However, such techniques have a worst-case complexity that scales exponentially in the number of decision variables, which severely limits applicability to only low-dimensional problems [5]. Also, in cases where the load distribution is unknown, the advanced forecasting models embedded within stochastic optimization techniques [6], [7] also incur high computational burden. In contrast, reinforcement learning (RL) approaches for stochastic dynamic optimization problems have been shown to scale gracefully in complex, high-dimensional environments, even without knowledge of the load distribution [8]. Furthermore, standard stochastic optimization-based solutions rely on a *centralized* approach for both forecasting and optimizing. However, in interconnected power market operation, participants may not willingly disclose their proprietary forecasts for spatio-temporal intermittent generation and/or net-load uncertainty because these have significant economic value. In this context, decentralized approaches for learning to optimize (or to clear a market) by agents while ensuring forecast information remains *private* are compelling directions for research.

To address the aforementioned shortcomings, in this work, we develop a decentralized multi-agent reinforcement learning (MARL) algorithm for scheduling power and interconnection flows in a decentralized fashion. The approach is based on an *actor-critic* algorithm which has shown to be quite successful in solving high-dimensional problems [8]. Via numerical simulations involving a three-area test system, we demonstrate that the proposed scheme minimizes expected net operation costs over multiple time periods without requiring centralized forecasting or coordination. We show that resulting RL policies generalize to the test data sampled independently of the training data and adapt to system uncertainty.

The remainder of this paper is organized as follows. In Section II, we provide a review of reinforcement learning applications in power system operation. We outline the multi-area power scheduling problem with uncertainty in Section III. In Section IV, we introduce a bi-level formulation for the scheduling problem under study and propose a decentralized multi-agent actor-critic algorithm to solve it. We evaluate the proposed algorithm in a toy example involving a three-area

Submitted to the 23rd Power Systems Computation Conference (PSCC 2024).

test system and compare the performance with those of model predictive control (MPC) and centralized reinforcement learning as benchmark techniques in Section V. Finally, Section VI offers concluding remarks.

II. Literature Review

To mitigate the computational burden arising from increased system size and to preserve the privacy of areas potentially operating under different entities, distributed optimization techniques for power system operation have received prominent attention [9]. Networked MARL offers a decentralized approach for addressing complex, large-scale control challenges. This method allows communication among neighboring agents, enabling its application in sectors such as traffic management, as highlighted in [10]. [11] advances this field by providing convergence guarantees within a linear function approximation framework. Additionally, the proposed model in [12] involves the use of deep neural networks to encode state information, along with the construction of critics and actors for agents, aiming to maximize a weighted return of rewards within a close neighborhood. Despite the potential of networked MARL to learn optimal and stable policies, it requires global information about either the state or the reward. [12], [13]. In what follows we briefly review a growing body of literature on MARL approaches for power system operation.

A deep deterministic policy gradient in a multi-agent deep RL framework is employed in [14] to solve the load frequency control problem. This method requires significant computational resources and is designed to operate in environments without uncertainty. In [15], the authors used a deep Q-network for the distributed nonconvex economic dispatch problem, an approach effective yet constrained by the time needed for system-wide consensus which leads to a heavy communication burden. A hierarchical RL method is used in [16] to solve the multi-area economic dispatch problem. This methodology bears similarities with our work with a two-layer problem decomposition. The bottom layer uses Q-learning for independent economic dispatch problems in each area, and the top layer optimizes power flows on tielines. However, the purpose of the usage of Q-learning is to determine the optimal power outputs in a sequential manner rather than solve a multi-stage problem with multiple time periods as we do. Like other approaches [17]–[23] implementing RL to the power system problem, [14]–[16] use discrete action space (e.g., generator outputs, transmission line flows), which is accompanied by a trade-off between computational cost and solution quality. Finer granularity in the action space requires more computational effort during training but typically yields higher-quality solutions. In contrast, the agents in our proposed algorithm employ Gaussian policies to identify the optimal decisions directly within the continuous action spaces.

III. Multi-area Power Scheduling Problem

In this section, we introduce notation, models, and operational constraints pertinent to nodal loads, generators, energy storage devices, transmission lines, and tielines. We also formulate the multi-area power scheduling problem.

A. Notation and Operational Constraints

Consider an interconnected power network composed of A areas indexed in the set $\mathcal{A} = \{1, \ldots, A\}$, where each area $a \in \mathcal{A}$ is described by a directed graph $(\mathcal{N}_a, \mathcal{L}_a)$ with $\mathcal{N}_a = \{1, \ldots, N_a\}$ representing the set of buses and $\mathcal{L}_a = \{(i,j) | i, j \in \mathcal{N}_a, j \equiv j(i)\}$ representing the set of transmission lines therein (lines inside areas). We further collect the tielines (lines between areas) in the set $\mathcal{L}^{\text{tie}} = \{(i,j) | i \in \mathcal{N}_a, j \in \mathcal{N}_{a'}, j \equiv j(i), a, a' \in \mathcal{A}\}$. Our focus is a multi-stage scheduling problem comprising T time periods collected in $\mathcal{T} = \{1, \ldots, T\}$, and the goal is to maximize the cumulative gains from power exchange amongst areas. Nodal loads in each area $a \in \mathcal{A}$ are supplied by area generation fleet composed of G_a online generators indexed in the set $\mathcal{G}_a = \{1, \ldots, G_a\}$, K_a energy storage devices indexed in $\mathcal{K}_a = \{1, \ldots, K_a\}$, and possibly interarea tieline inflow.

1) Nodal Loads: At time $t \in \mathcal{T}$, nodal loads in area $a \in \mathcal{A}$ are decomposed into three components: i) an inflexible component $\mathbf{L}^{\text{i}}_{a,t} = [(L^{\text{i}}_{n,a,t})_{n \in \mathcal{N}_a}]^\top$ requiring immediate fulfillment, ii) a flexible component $\mathbf{L}^{\text{f}}_{a,t} = [(L^{\text{f}}_{n,a,t})_{n \in \mathcal{N}_a}]^\top$ allowing for deferred fulfillment, and iii) an elastic component $\mathbf{L}^{\text{e}}_{a,t} = [(L^{\text{e}}_{n,a,t})_{n \in \mathcal{N}_a}]^\top$ representing price-sensitive demand that can be adjusted to satisfy consumer utility.

Inflexible loads are subject to uncertainty $\epsilon_{a,t} = [(\epsilon_{n,a,t})_{n \in \mathcal{N}_a}]^\top$. Uncertainty here can arise due to errors in day-ahead forecasts of loads and must-run renewable energy sources, the latter mainly the result of the unpredictable nature of the primary source of energy, e.g., solar radiation or wind speed. Commonly used distributions to model the uncertainty include Gaussian, Beta [24], Gamma [25], Weibull [26], and lognormal [27] distributions. However, in practice, these explicit distributions may be difficult to obtain, or they may yield suboptimal or conservative decisions. Moreover, since accurate forecast distributions have significant commercial value, individual market participants may not be willing to disclose such information to a central decision maker.

For flexible loads, denote the deferred amount (i.e., the difference between the desired load $\mathbf{L}^{\text{f}}_{a,t}$ and the scheduled load at time t) by $\mathbf{L}^{\text{d}}_{a,t} = [(L^{\text{d}}_{n,a,t})_{n \in \mathcal{N}_a}]^\top$. Negative-valued entries in $\mathbf{L}^{\text{d}}_{a,t}$ represent previously deferred load supplied at time t, and nonnegative-valued entries therein represent the amount deferred at time t. We further model the sum of deferred loads up to time t (the queued load) with the positive-valued variable $\mathbf{D}_{a,t} = [(D_{n,a,t})_{n \in \mathcal{N}_a}]^\top$, governed by the following state equation:

$$\mathbf{0} \leq \mathbf{D}_{a,t+1} = \mathbf{D}_{a,t} + \mathbf{L}^{\text{d}}_{a,t}, \ a \in \mathcal{A}, \ t \in \mathcal{T} \backslash \{T\}, \qquad (1)$$

where its initial value is

$$\mathbf{D}_{a,1} = \mathbf{0}, \ a \in \mathcal{A}. \qquad (2)$$

Further, the condition below ensures that all deferred loads are supplied by the end of scheduling horizon

$$\mathbf{D}_{a,T} = \mathbf{0}, \ a \in \mathcal{A}. \qquad (3)$$

As per its definition, $\mathbf{L}^{\mathrm{d}}_{a,t}$ is constrained as follows:

$$-\mathbf{D}_{a,t} \leq \mathbf{L}^{\mathrm{d}}_{a,t} \leq \mathbf{L}^{\mathrm{f}}_{a,t},\ a \in \mathcal{A},\ t \in \mathcal{T}. \tag{4}$$

Finally, the third component of nodal loads, i.e., elastic loads, is nonnegative in value and it is constrained with lower and upper limits as[1]

$$\mathbf{0} \leq \mathbf{L}^{\mathrm{e}}_{a,t} \leq \overline{\mathbf{L}}^{\mathrm{e}}_{a},\ a \in \mathcal{A},\ t \in \mathcal{T}. \tag{5}$$

2) Generators: Denote the power produced by generators by $\mathbf{P}_{a,t} = [(P_{g,a,t})_{g \in \mathcal{G}_a}]^\top$. Power output limits of generators are imposed through the following upper and lower bounds:

$$\underline{\mathbf{P}}_a \leq \mathbf{P}_{a,t} \leq \overline{\mathbf{P}}_a,\ a \in \mathcal{A},\ t \in \mathcal{T}. \tag{6}$$

In the interest of simplicity, we do not include generator ramping limits, but we note that they can be incorporated in a straightforward manner at the expense of greater notational and computational burden.

3) Energy Storage Devices: Denote the charging and discharging power of energy storage devices respectively by $\mathbf{P}^{\mathrm{c}}_{a,t} = [(P^{\mathrm{c}}_{k,a,t})_{k \in \mathcal{K}_a}]^\top$ and $\mathbf{P}^{\mathrm{d}}_{a,t} = [(P^{\mathrm{d}}_{k,a,t})_{k \in \mathcal{K}_a}]^\top$, and their energy by $\mathbf{E}_{a,t} = [(E_{k,a,t})_{k \in \mathcal{K}_a}]^\top$. The energy storage charging and discharging power limits are enforced as

$$\mathbf{0} \leq \mathbf{P}^{\mathrm{c}}_{a,t} \leq \overline{\mathbf{P}}^{\mathrm{c}}_{a},\ a \in \mathcal{A},\ t \in \mathcal{T}, \tag{7a}$$

$$\mathbf{0} \leq \mathbf{P}^{\mathrm{d}}_{a,t} \leq \overline{\mathbf{P}}^{\mathrm{d}}_{a},\ a \in \mathcal{A},\ t \in \mathcal{T}, \tag{7b}$$

and the stored energy is calculated as

$$\mathbf{E}_{a,t+1} = \mathbf{E}_{a,t} + \boldsymbol{\eta}^{\mathrm{c}}_a \mathbf{P}^{\mathrm{c}}_{a,t} - {\boldsymbol{\eta}^{\mathrm{d}}_a}^{-1} \mathbf{P}^{\mathrm{d}}_{a,t}, a \in \mathcal{A}, t \in \mathcal{T} \backslash \{T\}, \tag{8}$$

where $\boldsymbol{\eta}^{\mathrm{c}}_a$ and $\boldsymbol{\eta}^{\mathrm{d}}_a$ respectively represent diagonal matrices of charging and discharging efficiencies, and with initial condition

$$\mathbf{E}_{a,1} = \mathbf{E}^{\mathrm{ini}}_a,\ a \in \mathcal{A}. \tag{9}$$

The amount of stored energy is constrained to

$$\underline{\mathbf{E}}_a \leq \mathbf{E}_{a,t} \leq \overline{\mathbf{E}}_a,\ a \in \mathcal{A},\ t \in \mathcal{T}. \tag{10}$$

4) Power Deficits: In the event of a power shortfall, the power deficit $\mathbf{L}^{\mathrm{dfc}}_{a,t} = [(L^{\mathrm{dfc}}_{n,a,t})_{n \in \mathcal{N}_a}]^\top$ would be procured from more expensive emergency sources, and it is confined to

$$\mathbf{0} \leq \mathbf{L}^{\mathrm{dfc}}_{a,t} \leq \mathbf{L}^{\mathrm{i}}_{a,t},\ a \in \mathcal{A},\ t \in \mathcal{T}. \tag{11}$$

5) Transmission Lines and Tielines: Denote intra-area transmission line and inter-area tieline power flows with $\mathbf{F}_{a,t} = [(F_{(\imath,\jmath),a,t})_{(\imath,\jmath) \in \mathcal{L}_a}]^\top$ and $\mathbf{T}_t = [(T_{(\imath,\jmath),t})_{(\imath,\jmath) \in \mathcal{L}^{\mathrm{tie}}}]^\top$, respectively. Transmission line and tieline power flows are constrained to their thermal limits as

$$-\overline{\mathbf{F}}_a \leq \mathbf{F}_{a,t} \leq \overline{\mathbf{F}}_a,\ a \in \mathcal{A},\ t \in \mathcal{T}, \tag{12a}$$

$$-\overline{\mathbf{T}} \leq \mathbf{T}_t \leq \overline{\mathbf{T}},\ t \in \mathcal{T}. \tag{12b}$$

6) Nodal Power Balance: The nodal power balance is enforced through following constraint:

$$\begin{aligned}&\mathbf{M}^{\mathrm{g}}_a \mathbf{P}_{a,t} + \mathbf{M}^{\mathrm{es}}_a (\mathbf{P}^{\mathrm{d}}_{a,t} - \mathbf{P}^{\mathrm{c}}_{a,t}) - (\mathbf{L}^{\mathrm{i}}_{a,t} + \boldsymbol{\epsilon}_{a,t}) - \mathbf{L}^{\mathrm{e}}_{a,t} - \mathbf{L}^{\mathrm{f}}_{a,t} \\ &+ \mathbf{L}^{\mathrm{d}}_{a,t} + \mathbf{L}^{\mathrm{dfc}}_{a,t} - \mathbf{M}^{\mathrm{tr}}_a \mathbf{F}_{a,t} - \mathbf{M}^{\mathrm{tie}}_a \mathbf{T}_t = \mathbf{0}, a \in \mathcal{A}, t \in \mathcal{T},\end{aligned} \tag{13}$$

where $\mathbf{M}^{\mathrm{g}}_a \in \mathbb{R}^{G_a \times N_a}$ and $\mathbf{M}^{\mathrm{es}}_a \in \mathbb{R}^{K_a \times N_a}$ denote mappings respectively from generators and energy storage devices to buses, while $\mathbf{M}^{\mathrm{tr}}_a \in \mathbb{R}^{|\mathcal{L}_a| \times N_a}$ and $\mathbf{M}^{\mathrm{tie}}_a \in \mathbb{R}^{|\mathcal{L}^{\mathrm{tie}}| \times N_a}$ project respectively the transmission lines and tielines to buses. It is worth noting that the nodal power balance above uses line power flows directly (instead of relating them to nodal voltages as is commonly done in power systems literature). Particularly, flows collected in $\mathbf{F}_{a,t}$ and $\mathbf{T}_t$ will emerge as decision variables in the problem formulated next.

B. Problem Formulation

Let $C_a(\mathbf{P}_{a,t}, \mathbf{L}^{\mathrm{e}}_{a,t}, \mathbf{L}^{\mathrm{dfc}}_{a,t})$ represent the net operation cost for area a (as a quadratic function) including the power generation and power deficit procurement costs, less the elastic load surplus. For the sake of brevity, we use $\mathbf{X}_{a,t}$ to denote the local decisions of area $a \in \mathcal{A}$ at time $t \in \mathcal{T}$, i.e., $\mathbf{X}_{a,t} = [\mathbf{P}^\top_{a,t}, \mathbf{P}^{\mathrm{c}\top}_{a,t}, \mathbf{P}^{\mathrm{d}\top}_{a,t}, \mathbf{E}^\top_{a,t}, \mathbf{L}^{\mathrm{e}\top}_{a,t}, \mathbf{L}^{\mathrm{d}\top}_{a,t}, \mathbf{L}^{\mathrm{dfc}\top}_{a,t}, \mathbf{D}^\top_{a,t}, \mathbf{F}^\top_{a,t}]^\top$. Given that, the multi-area power scheduling problem is formulated as follows:

$$\min_{\mathbf{X}_{a,t}, \mathbf{T}_t} \ \mathbb{E}\Big[\sum_{a \in \mathcal{A}} \sum_{t \in \mathcal{T}} C_a(\mathbf{P}_{a,t}, \mathbf{L}^{\mathrm{e}}_{a,t}, \mathbf{L}^{\mathrm{dfc}}_{a,t})\Big] \tag{14a}$$

$$\text{s.t. } h(\mathbf{X}_{a,t}, \mathbf{T}_t) \leq \mathbf{0}, \qquad a \in \mathcal{A},\ t \in \mathcal{T}, \tag{14b}$$

$$g(\mathbf{X}_{a,t}, \mathbf{X}_{a,t+1}) = \mathbf{0}, \quad a \in \mathcal{A},\ t \in \mathcal{T} \backslash \{T\}, \tag{14c}$$

$$f(\mathbf{X}_{a,t}, \mathbf{T}_t, \mathbf{L}^{\mathrm{i}}_{a,t}, \mathbf{L}^{\mathrm{f}}_{a,t}, \boldsymbol{\epsilon}_{a,t}) = \mathbf{0}, a \in \mathcal{A}, t \in \mathcal{T}, \tag{14d}$$

where $\mathbb{E}[\,\cdot\,]$ denotes the expectation taken with respect to the uncertainty $\boldsymbol{\epsilon}_{a,t}$, (14b) collects the inequality constraints modeling the operational limits in (4)–(7) and (10)–(12), (14c) consists of equality constraints including those related to the dynamics in flexible loads in (1)–(3) and in energy storage devices in (8)–(9), and (14d) serves to compactly express the nodal power balance in (13).

C. Motivation for a MARL Solution Approach

The problem formulated in (14) is a convex quadratic optimization problem. As such, the problem is amenable to a centralized solution approach, assuming that the perturbations encapsulated by $\boldsymbol{\epsilon}_{a,t}$ are known for all $a \in \mathcal{A}$ and for all $t \in \mathcal{T}$. However, assuming perturbations $\boldsymbol{\epsilon}_{a,t}$ are precisely known a day ahead may be impractical in real-world scenarios since most perturbations occur in real time. On the other hand, if a forecast probability distribution of perturbations were available, the problem in (14) can be tackled with popular stochastic programming techniques such as stochastic dual decomposition programming (SDDP) [4]. However, SDDP has a worst-case complexity that scales exponentially in the number of decision variables [5]. This severely limits applicability to only low-dimensional problems and renders the computational cost of solving the problem in (14) for large-scale systems prohibitive for real-time decision making. Another relevant

[1] As a matter of convention, in the remainder of the paper, overlined and underlined variables respectively refer to their maximum and minimum limits.

facet of the problem in (14) pertains to information and privacy. As it is presented, the problem in (14) necessitates a central decision maker to possess comprehensive information from all buses in all areas. Such a structure falls short in providing privacy for each area and flexibility as a whole.

To address the aforementioned shortcomings, we develop a decentralized RL framework to optimally schedule the independent areas under the more realistic assumption that the perturbations are known only one hour ahead, which in turn leads to adopting a dynamic decision-making scheme. Particularly, RL approaches have been shown to scale gracefully in solving complex, high-dimensional stochastic dynamic optimization problems [8].

IV. Decentralized Multi-Agent Actor-Critic Solution Approach

In this section, we design a decentralized MARL algorithm to effectively and efficiently solve the problem in (14). The RL approach is inherently well suited for problems that involve sequential decision-making and real-time adjustment. Consequently, the agent can make timely decisions in each time period, responding to uncertain perturbations as they occur real time.

A. Bi-level Formulation

To offer a solution to the multi-area power exchange problem, we leverage a MARL framework that consists of two types of agents: *operating agents* and *interconnection agents*. In each area $a \in \mathcal{A}$, there exists a single operating agent responsible for making local decisions $\mathbf{X}_{a,t}$ that directly impact the net operational cost in that area, but it does not directly influence other areas. Further, for each tieline $(i, j) \in \mathcal{L}^{\text{tie}}$, there is an interconnection agent responsible for determining the power flows $T_{(i,j),t}$ using information from only bus i and bus j located in neighboring areas. As shown in Fig. 1, the operating (interconnection) agent shares limited information only with its neighboring interconnection (operating) agents, rather than submitting all detailed information to a central decision maker. Furthermore, a pair of neighboring operating and interconnection agents can collaborate effectively to handle perturbations without needing to solve a comprehensive multi-stage stochastic programming problem for the entire system. The decentralized multi-agent framework can be considered a bi-level problem. The lower-level problem involves training the operating agent in each area a to make optimal local decisions. Meanwhile, the upper-level problem focuses on training the interconnection agents to determine optimal tieline flows. This setting is akin to a multi-leader-multi-follower model within a cooperative context, where the interconnection agent indeed acts in a leading capacity, with its actions informing the subsequent decisions of the operating agents.

1) Lower-level Problem: The goal of the lower-level problem is to determine the optimal policy for local decisions. The operating agent in area $a \in \mathcal{A}$ can communicate with its neighboring interconnection agents, allowing it to acquire information regarding the tieline flows. With a minor abuse of notation, we use $\mathbf{M}_a^{\text{tie}}\mathbf{T}_t$ to denote the net power leaving the buses of area a through tielines, even when $\mathbf{T}_t$ is not fully available. This includes cases where the operating agent of area a is unaware of the flows on tielines connecting to areas other than a. To apply the RL approach, we define the Markov decision process (MDP) for each area a by a tuple $\mathcal{M}_a^{\text{lo}} = (\mathcal{S}_a^{\text{lo}}, \mathcal{U}_a^{\text{lo}}, R_a^{\text{lo}}, \mathcal{P}_a^{\text{lo}}, T)$, where $\mathcal{S}_a^{\text{lo}}$ denotes the state space and the state $\mathbf{s}_a^{\text{lo}} \in \mathcal{S}_a^{\text{lo}}$ is defined as $\mathbf{s}_{a,t}^{\text{lo}} = [\mathbf{L}_{a,t}^{\text{i}\top}, \mathbf{L}_{a,t}^{\text{f}\top}, \boldsymbol{\epsilon}_{a,t}^{\top}, \mathbf{D}_{a,t}^{\top}, \mathbf{E}_{a,t}^{\top}, (\mathbf{M}_a^{\text{tie}}\mathbf{T}_t)^{\top}]^{\top}$. Meanwhile, $\mathcal{U}_a^{\text{lo}}$ represents the action space that consists of all possible values of local action $\mathbf{u}_{a,t}^{\text{lo}} = [\mathbf{P}_{a,t}^{\top}, \mathbf{P}_{a,t}^{\text{c}\top}, \mathbf{P}_{a,t}^{\text{d}\top}, \mathbf{L}_{a,t}^{\text{e}\top}, \mathbf{L}_{a,t}^{\text{d}\top}, \mathbf{L}_{a,t}^{\text{dfc}\top}, \mathbf{F}_{a,t}^{\top}]^{\top}$. The area a local reward function is represented by R_a^{lo}, which consists of not only the net operational cost $C_a(\mathbf{P}_{a,t}, \mathbf{L}_{a,t}^{\text{e}}, \mathbf{L}_{a,t}^{\text{dfc}})$ but also penalties for actions that violate the constraints in (14). The dynamics $\mathcal{P}_a^{\text{lo}}$ specify how the state changes over time, i.e., $\mathbf{s}_{a,t+1}^{\text{lo}} \sim \mathcal{P}_{a,t}^{\text{lo}}(\cdot \mid \mathbf{s}_{a,t}^{\text{lo}}, \mathbf{u}_{a,t}^{\text{lo}})$. Particularly, $\mathbf{L}_{a,t+1}^{\text{i}}$ and $\mathbf{L}_{a,t+1}^{\text{f}}$ are determined by day-ahead predictions; $\boldsymbol{\epsilon}_{a,t+1}$ follows a particular distribution which is unknown to the agent and may depend on $\boldsymbol{\epsilon}_{a,t}$; $\mathbf{D}_{a,t+1}$ and $\mathbf{E}_{a,t+1}$ are updated according to (1) and (8), respectively; and $\mathbf{M}_a^{\text{tie}}\mathbf{T}_{t+1}$ depends on the policies of neighboring interconnection agents. The final element of $\mathcal{M}_a^{\text{lo}}$ signifies that this MDP spans T time periods. It is worth emphasizing that the local decisions made in one area affect the MDPs of other areas *indirectly* only through the neighboring interconnection agents. This implies that the lower-level problems can be solved independently for each area in a decentralized manner (without direct communication among areas). Graphical illustration of the MDP of the lower-level problem is presented in Fig. 2.

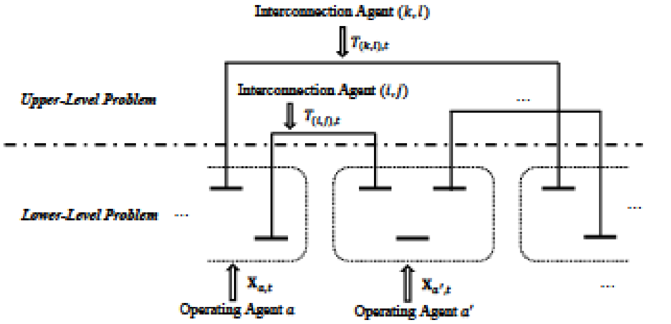

Fig. 1. Structure of bi-level multi-agent reinforcement learning. In the upper-level problem, interconnection agents aim to determine optimal tieline flows, while in the lower-level problem, operating agents intend to optimize the net operational cost for each area given these tieline flows.

In each area a, the policy π_a^{lo} of the operating agent actor consists of the choice probability $\pi_{a,t}^{\text{lo}}(\cdot \mid \mathbf{s}_{a,t}^{\text{lo}}) = \mathcal{N}(\mu_{a,t}^{\text{lo}}(\mathbf{s}_{a,t}^{\text{lo}}), {\sigma_{a,t}^{\text{lo}}}^2)$ for $t \in \mathcal{T}$, $\mathbf{s}_{a,t}^{\text{lo}} \in \mathcal{S}_a^{\text{lo}}$, where the mean function $\mu_{a,t}^{\text{lo}}(\cdot)$ is parameterized by the weight $\phi_{a,t}^{\text{lo}}$ and the variance ${\sigma_{a,t}^{\text{lo}}}^2$ is considered as a hyperparameter. This strategy has the potential to decrease the training burden. Furthermore, as the variance governs the exploration-exploitation trade-off of the policy, it is advisable to start with a large value and

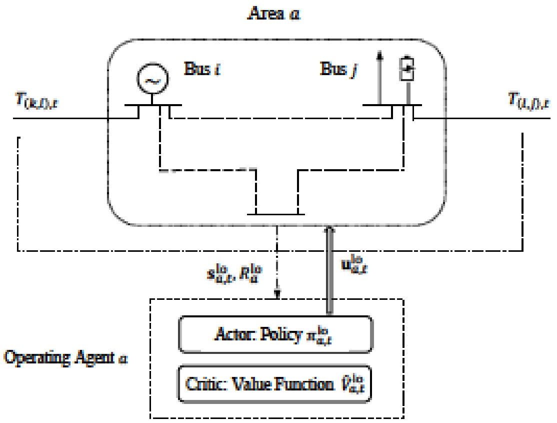
Area a
Bus i
Bus j
$T_{(k,i),t}$
$T_{(i,j),t}$
$\mathbf{s}^{\mathrm{lo}}_{a,t}, R^{\mathrm{lo}}_{a}$
$\mathbf{u}^{\mathrm{lo}}_{a,t}$
Actor: Policy $\pi^{\mathrm{lo}}_{a,t}$
Critic: Value Function $\widehat{V}^{\mathrm{lo}}_{a,t}$
Operating Agent a

Fig. 2. Illustration of MDP for lower-level problem. Operating agent for area $a \in \mathcal{A}$ is responsible for making local decisions given the information about area a and neighboring tieline flows.

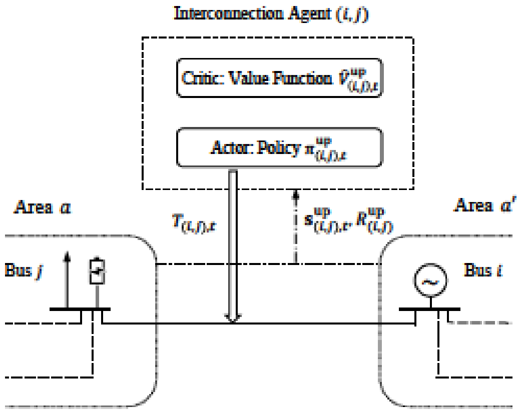
Interconnection Agent (i,j)
Critic: Value Function $\widehat{V}^{\mathrm{up}}_{(i,j),t}$
Actor: Policy $\pi^{\mathrm{up}}_{(i,j),t}$
Area a
$T_{(i,j),t}$
$\mathbf{s}^{\mathrm{up}}_{(i,j),t}, R^{\mathrm{up}}_{(i,j)}$
Area a'
Bus j
Bus i

Fig. 3. Illustration of MDP for upper-level problem. Interconnection agent for tieline $(i,j) \in \mathcal{L}_{\mathrm{tie}}$ determines the power flows on the tieline (i,j) given the information from buses $i \in \mathcal{N}_a$ and $j \in \mathcal{N}_{a'}$, $a, a' \in \mathcal{A}$.

gradually decrease it during the learning process intentionally. The value function is approximated by the critic as $\widehat{V}^{\mathrm{lo}}_{a,t}$ with parameter $\omega^{\mathrm{lo}}_{a,t}$. This approximation represents the expected sum of future rewards realized by employing π^{lo}_{a}. For area $a \in \mathcal{A}$, the actor-critic algorithm updates the parameters of the operating agent, given the transition tuple $(\mathbf{s}^{\mathrm{lo}}_{a,t}, \mathbf{u}^{\mathrm{lo}}_{a,t}, \mathbf{s}^{\mathrm{lo}}_{a,t+1})$, as follows:

$$\begin{aligned}\mathrm{TD}^{\mathrm{lo}}_{a,t} \leftarrow {} & R^{\mathrm{lo}}_{a}(\mathbf{s}^{\mathrm{lo}}_{a,t}, \mathbf{u}^{\mathrm{lo}}_{a,t}) + \widehat{V}^{\mathrm{lo}}_{a,t+1}(\mathbf{s}^{\mathrm{lo}}_{a,t+1}) \\ & - \widehat{V}^{\mathrm{lo}}_{a,t}(\mathbf{s}^{\mathrm{lo}}_{a,t}), \qquad (15a)\\ \phi^{\mathrm{lo}}_{a,t} \leftarrow {} & \phi^{\mathrm{lo}}_{a,t} + \beta^{\mathrm{lo,ac}} \cdot \mathrm{TD}^{\mathrm{lo}}_{a,t} \\ & \cdot \nabla_{\phi^{\mathrm{lo}}_{a,t}} \log \pi^{\mathrm{lo}}_{a,t}(\mathbf{u}^{\mathrm{lo}}_{a,t} \mid \mathbf{s}^{\mathrm{lo}}_{a,t}), \qquad (15b)\\ \omega^{\mathrm{lo}}_{a,t} \leftarrow {} & \omega^{\mathrm{lo}}_{a,t} + \beta^{\mathrm{lo,cr}} \cdot \mathrm{TD}^{\mathrm{lo}}_{a,t} \cdot \nabla_{\omega^{\mathrm{lo}}_{a,t}} \widehat{V}^{\mathrm{lo}}_{a,t}(\mathbf{s}^{\mathrm{lo}}_{a,t}), \qquad (15c)\end{aligned}$$

where $\beta^{\mathrm{lo,ac}}$ and $\beta^{\mathrm{lo,cr}}$ respectively represent the learning rates of the actor and critic associated with the operating agent. Note that $\widehat{V}^{\mathrm{lo}}_{a,T+1}(s) = 0, \forall s \in \mathcal{S}^{\mathrm{lo}}_{a}$ by convention.

To ensure the feasibility of the actions, we apply projection and clipping steps. For instance, for the decision $\mathbf{P}^{\mathrm{c}}_{a,t}$, we initially project the raw value from the actor into the range $[\mathbf{0}, \overline{\mathbf{P}}^{\mathrm{c}}_{a}]$ to obtain $\mathbf{P}^{\mathrm{c}}_{a,t,\mathrm{proj}}$. Then, if the energy $\mathbf{E}_{a,t} + \boldsymbol{\eta}^{\mathrm{c}}_{a}\mathbf{P}^{\mathrm{c}}_{a,t,\mathrm{proj}}$ exceeds the capacity $\overline{\mathbf{E}}_{a}$, we clip the value. The final adjusted value, $\mathbf{P}^{\mathrm{c}}_{a,t}$, is the element-wise minimum between $\mathbf{P}^{\mathrm{c}}_{a,t,\mathrm{proj}}$ and $\boldsymbol{\eta}^{\mathrm{c}\,-1}_{a}(\overline{\mathbf{E}}_{a} - \mathbf{E}_{a,t})$.

The inequality $h(\mathbf{X}_{a,t}, \mathbf{T}_t)) \leq 0$ is handled by a primal-dual approach wherein a penalty term is defined as

$$-\mathbf{y}^{\top}_{a,t} \max(\mathbf{0}, h(\mathbf{X}_{a,t,\mathrm{proj}}, \mathbf{T}_t))$$

with $\mathbf{y}_{a,t} \geq 0$ representing the vector of Lagrange multipliers updated in every iteration and $\mathbf{X}_{a,t,\mathrm{proj}}$ indicating the action before the clipping step.

2) Upper-level Problem: Given the presence of perturbed loads, fixed tieline flows may not be optimal for all scenarios. To effectively address this uncertainty, interconnection agents need to learn policies that adapt the flows on tielines to handle the perturbations. Therefore, we formulate the interconnection agent problem (upper level) under the MDP framework and solve it using the actor-critic algorithm. For each tieline $(i,j) \in \mathcal{L}^{\mathrm{tie}}$ connecting buses $i \in \mathcal{N}_a$ and $j \in \mathcal{N}_{a'}$, $a, a' \in \mathcal{A}$, we define an MDP $\mathcal{M}^{\mathrm{up}}_{(i,j)} = (\mathcal{S}^{\mathrm{up}}_{(i,j)}, \mathcal{U}^{\mathrm{up}}_{(i,j)}, R^{\mathrm{up}}_{(i,j)}, \mathcal{P}^{\mathrm{up}}_{(i,j)}, T)$, where $\mathcal{S}^{\mathrm{up}}_{(i,j)}$ is the state space that comprises all possible values of the inflexible loads, flexible loads, perturbations, queued flexible loads, and stored energy at tieline end buses i and j. Thereby, the state $\mathbf{s}^{\mathrm{up}}_{(i,j),t}$ is formed as $\mathbf{s}^{\mathrm{up}}_{(i,j),t} = [\hat{\mathbf{s}}^{\mathrm{lo}\,\top}_{i,t}, \hat{\mathbf{s}}^{\mathrm{lo}\,\top}_{j,t}]^{\top}$, where $\hat{\mathbf{s}}^{\mathrm{lo}}_{i,t} = [L^{\mathrm{i}}_{i,a,t}, L^{\mathrm{f}}_{i,a,t}, \epsilon_{i,a,t}, D_{i,a,t}, \mathbf{M}^{\mathrm{es}}_{i,a}\mathbf{E}_{a,t}]^{\top}$ is the partial state for the bus i and $\mathbf{M}^{\mathrm{es}}_{i,a}$ is the i-th row of $\mathbf{M}^{\mathrm{es}}_{a}$. The action space $\mathcal{U}^{\mathrm{up}}_{(i,j)}$ consists of all possible values of tieline power flows $T_{(i,j),t}$ between buses i and j at time t. Since the flow is only constrained by (12b), we can ensure the feasibility of the action $T_{(i,j),t}$ by projecting it onto the feasible range. A reward function of $R^{\mathrm{up}}_{(i,j)}$ governs the decisions of the agent associated with tieline $(i,j) \in \mathcal{T}^{\mathrm{tie}}$. Under the assumption that each tieline end bus accommodates at least one generator or elastic load, we define the *marginal price* at the boundary bus $i \in \mathcal{N}_a$ as

$$\alpha_i\left(\mathbf{u}^{\mathrm{lo}}_{a,t}\right) = \begin{cases} \partial C_a(\mathbf{u}^{\mathrm{lo}}_{a,t}) / \partial(\mathbf{M}^{\mathrm{g}}_{i,a}\mathbf{P}_{a,t}), & \text{if } \mathbf{M}^{\mathrm{g}}_{i,a} \neq \mathbf{0} \\ & \text{and } L^{\mathrm{dfc}}_{i,a,t} = 0, \\ \partial C_a(\mathbf{u}^{\mathrm{lo}}_{a,t}) / \partial L^{\mathrm{e}}_{i,a,t}, & \text{if } \mathbf{M}^{\mathrm{g}}_{i,a} = \mathbf{0} \\ & \text{and } L^{\mathrm{dfc}}_{i,a,t} = 0, \\ \partial C_a(\mathbf{u}^{\mathrm{lo}}_{a,t}) / \partial L^{\mathrm{dfc}}_{i,a,t}, & \text{otherwise,} \end{cases} \tag{16}$$

where $\mathbf{u}^{\mathrm{lo}}_{a,t}$ and $\mathbf{u}^{\mathrm{lo}}_{a',t}$ are the actions made by operating agents at lower level, and $\mathbf{M}^{\mathrm{g}}_{i,a}$ is the i-th row of $\mathbf{M}^{\mathrm{g}}_{a}$. The marginal price in electricity markets indicates the cost of producing one more unit of electricity, the revenue from supplying that additional unit of elastic load, or the expense of obtaining one unit from an emergency source during a power deficit. The optimal flows can be determined by minimizing the discrepancy between the marginal prices of the two interconnected boundary buses. Therefore, the reward is formulated as

$$\begin{aligned} R^{\mathrm{up}}_{(i,j)}(\mathbf{s}^{\mathrm{up}}_{(i,j),t}, T_{(i,j),t}) = -\left(\alpha_i\left(\mathbf{u}^{\mathrm{lo}}_{a,t}\right) - \alpha_j\left(\mathbf{u}^{\mathrm{lo}}_{a',t}\right)\right)^2, \\ \forall (i,j) \in \mathcal{L}^{\mathrm{tie}},\ t \in \mathcal{T}. \end{aligned} \tag{17}$$

For interconnection agents, the policies adopted by the operating agents are viewed as integral parts of the environment. This is because they can influence both the reward and the dynamics $\mathcal{P}^{\text{up}}_{(i,j)}$. Graphical illustration of the MDP of the upper-level problem is presented in Fig. 3.

Similar to the actor-critic algorithm in the lower-level problem, the approximated value function attributed to tieline $(i,j) \in \mathcal{L}^{\text{tie}}$, denoted by $\widehat{V}^{\text{up}}_{(i,j),t}$, is parameterized via $\omega^{\text{up}}_{(i,j),t}$ while the mean function $\mu^{\text{up}}_{(i,j),t}$ of the corresponding actor's policy is parameterized by $\phi^{\text{up}}_{(i,j),t}$. Given the transition tuple $(\mathbf{s}^{\text{up}}_{(i,j),t}, T_{(i,j),t}, \mathbf{s}^{\text{up}}_{(i,j),t+1})$ for tieline (i,j), the parameters for the actor and critic are updated as follows:

$$\begin{aligned} \text{TD}^{\text{up}}_{(i,j),t} \leftarrow{}& R^{\text{up}}_{(i,j)}(\mathbf{s}^{\text{up}}_{(i,j),t}, T_{(i,j),t}) \\ &+ \widehat{V}^{\text{up}}_{(i,j),t+1}(\mathbf{s}^{\text{up}}_{(i,j),t+1}) - \widehat{V}^{\text{up}}_{(i,j),t}(\mathbf{s}^{\text{up}}_{(i,j),t}), \quad \text{(18a)} \\ \phi^{\text{up}}_{(i,j),t} \leftarrow{}& \phi^{\text{up}}_{(i,j),t} + \beta^{\text{up,ac}} \cdot \text{TD}^{\text{up}}_{(i,j),t} \\ &\cdot \nabla_{\phi^{\text{up}}_{(i,j),t}} \log \pi^{\text{up}}_{(i,j),t}(T_{(i,j),t} \mid \mathbf{s}^{\text{up}}_{(i,j),t}), \quad \text{(18b)} \\ \omega^{\text{up}}_{(i,j),t} \leftarrow{}& \omega^{\text{up}}_{(i,j),t} + \beta^{\text{up,cr}} \cdot \text{TD}^{\text{up}}_{(i,j),t} \\ &\cdot \nabla_{\omega^{\text{up}}_{(i,j),t}} \widehat{V}^{\text{up}}_{(i,j),t}(\mathbf{s}^{\text{up}}_{(i,j),t}), \quad \text{(18c)} \end{aligned}$$

where $\beta^{\text{up,ac}}$ and $\beta^{\text{up,cr}}$ are the learning rates for the interconnection actor and critic, respectively.

B. Proposed Algorithm

Our proposed bi-level decentralized multi-agent actor-critic algorithm is outlined in Algorithm 1. The operating agents undergo training phases that are interspersed with those of the interconnection agents, ensuring that both types of agents receive alternating periods of refinement. All Gaussian policies in the algorithm, associated with the operating and interconnection agents, have the same variance values, ${\sigma^{\text{lo}}}^2$ and ${\sigma^{\text{up}}}^2$, across all areas/tielines and time periods. Initially, these variances are set to be larger and are later manually decreased. This choice is driven by the intuition that we would like the policies to exhibit significant exploration during the initial phase and gradually converge to optimality.

Unlike typical RL implementations, the operating agents at lower level are learning to optimize a *non-stationary* environment. This is because the interconnection policies are changing every iteration. Similarly, the interconnection agents (upper level) are learning to maximize inter-temporal gains from trade between adjacent operating areas with nodal pricing policies that change every iteration. An intuitive explanation of why learning can take place in such *non-stationary* environment centers on the fact that *all* agents are implicitly pursuing the *same learning goal*, that is, to optimize the combined operation of all areas. To achieve stability, the algorithm uses a two-timescale update framework in which the policies of the interconnection agents are updated more *slowly* than those of the operating agents. This slower update rate allows the operating agents to approximately maximize surplus *within* their respective areas while the scheduled intertie flows change *slightly*. As a result, operating agents are able to learn the *local* operation that maximizes local surplus (lower level) and interconnection agents learn to maximize gains from trade between areas (upper level). We reserve the theoretical analysis of the convergence of the proposed scheme for future work.

In our implementation, the dynamics $\mathcal{P}^{\text{lo}}_{a}$ of area $a \in \mathcal{A}$ are not directly accessible. This is due to the fact that they are influenced by the policies of the interconnection agents. As a result, we employ an alternative method to sample the subsequent state $\mathbf{s}^{\text{lo}}_{a,t+1}$. We begin by defining the partial state of area $a \in \mathcal{A}$ at $t \in \mathcal{T}$ as $\vec{\mathbf{s}}^{\text{lo}}_{a,t} = [\mathbf{L}^{\text{i}\top}_{a,t}, \mathbf{L}^{\text{f}\top}_{a,t}, \boldsymbol{\epsilon}^{\top}_{a,t}, \mathbf{D}^{\top}_{a,t}, \mathbf{E}^{\top}_{a,t}]^{\top}$. Given this, the partial state $\vec{\mathbf{s}}^{\text{lo}}_{i,t}, \forall i \in \mathcal{N}_a$ can be derived from $\vec{\mathbf{s}}^{\text{lo}}_{a,t}$. The subsequent partial state $\vec{\mathbf{s}}^{\text{lo}}_{a,t+1}$ is drawn from the distribution $\widehat{\mathcal{P}}^{\text{lo}}_{a,t}(\cdot \mid \mathbf{s}^{\text{lo}}_{a,t}, \mathbf{u}^{\text{lo}}_{a,t})$. This distribution represents the transition probability $\mathcal{P}^{\text{lo}}_{a,t}$ marginalized over the tieline flows $\mathbf{M}^{\text{tie}}_{a}\mathbf{T}_{t+1}$ and is available. Therefore, in Algorithm 1, Step 12 is actually executed as detailed below:

$$\begin{aligned} \vec{\mathbf{s}}^{\text{lo}}_{a,t+1} &\sim \widehat{\mathcal{P}}^{\text{lo}}_{a,t}(\cdot \mid \mathbf{s}^{\text{lo}}_{a,t}, \mathbf{u}^{\text{lo}}_{a,t}),\ \forall a \in \mathcal{A}, && \text{(19a)} \\ \mathbf{s}^{\text{up}}_{(i,j),t+1} &= \left[\vec{\mathbf{s}}^{\text{lo}\,\top}_{i,t+1}, \vec{\mathbf{s}}^{\text{lo}\,\top}_{j,t+1}\right]^{\top},\ \forall (i,j) \in \mathcal{L}^{\text{tie}}, && \text{(19b)} \\ T_{(i,j),t+1} &= \mu^{\text{up}}_{(i,j),t+1}(\mathbf{s}^{\text{up}}_{(i,j),t+1}),\ \forall (i,j) \in \mathcal{L}^{\text{tie}}, && \text{(19c)} \\ \mathbf{s}^{\text{lo}}_{a,t+1} &= \left[\vec{\mathbf{s}}^{\text{lo}\,\top}_{a,t+1}, (\mathbf{M}^{\text{tie}}_{a}\mathbf{T}_{t+1})^{\top}\right]^{\top},\ \forall a \in \mathcal{A}. && \text{(19d)} \end{aligned}$$

We use the mean of the interconnection agent's policy as the action in (19c). Adopting deterministic actions from the interconnection agents results in a less stochastic environment for the operating agents, thereby aiding in stabilizing the training process.

Similarly, for tieline $(i,j) \in \mathcal{L}^{\text{tie}}$, we adopt an alternative method to replace the traditional process of sampling the subsequent state $\mathbf{s}^{\text{up}}_{(i,j),t+1}$ from the dynamics $\mathcal{P}^{\text{up}}_{(i,j)}$ with a more practical approach. In Algorithm 1, the procedure for Step 23 is elaborated as follows:

$$\begin{aligned} \mathbf{s}^{\text{lo}}_{a,t} &= \left[\vec{\mathbf{s}}^{\text{lo}\,\text{T}}_{a,t}, (\mathbf{M}^{\text{tie}}_{a}\mathbf{T}_{t})^{\top}\right]^{\top},\ \forall a \in \mathcal{A}, && \text{(20a)} \\ \mathbf{u}^{\text{lo}}_{a,t} &= \mu^{\text{lo}}_{a,t}(\mathbf{s}^{\text{lo}}_{a,t}),\ \forall a \in \mathcal{A}, && \text{(20b)} \\ \vec{\mathbf{s}}^{\text{lo}}_{a,t+1} &\sim \widehat{\mathcal{P}}^{\text{lo}}_{a,t}(\cdot \mid \mathbf{s}^{\text{lo}}_{a,t}, \mathbf{u}^{\text{lo}}_{a,t}),\ \forall a \in \mathcal{A}, && \text{(20c)} \\ \mathbf{s}^{\text{up}}_{(i,j),t+1} &= \left[\vec{\mathbf{s}}^{\text{lo}\,\text{T}}_{i,t+1}, \vec{\mathbf{s}}^{\text{lo}\,\text{T}}_{j,t+1}\right]^{\top},\ \forall (i,j) \in \mathcal{L}^{\text{tie}}. && \text{(20d)} \end{aligned}$$

V. Numerical Simulations

In this section, we present simulation results involving a three-area test system shown in Fig. 4. Results demonstrate the efficacy and efficiency of the proposed bi-level decentralized multi-agent actor-critic algorithm.

A. Simulation Setup

Here, we describe the simulation setup with respect to the test system, loads and perturbations, and parameters in the proposed algorithm.

1) Test System: We assess the performance of our proposed algorithm using a synthesized network comprising three interconnected areas, contained in set $\mathcal{A} = \{1,2,3\}$, with each area $a \in \mathcal{A}$ consisting of three buses $\mathcal{N}_a = \{1,2,3\}$, as shown in Fig. 4. Each of the three areas embeds three transmission lines (dashed traces) and each pair of areas are connected through two tielines (solid traces). The tielines form the set

Algorithm 1 Bi-Level Decentralized Multi-Agent Actor-Critic Algorithm

```
Require: Learning rates $\beta^{\text{lo,ac}}, \beta^{\text{lo,cr}}, \beta^{\text{up,ac}}, \beta^{\text{up,cr}}$; Initial
    variances of Gaussian policies ${\sigma^{\text{lo}}}^2, {\sigma^{\text{up}}}^2$; Variance reduction factors
    $\gamma^{\text{lo}}$, $\gamma^{\text{up}}$; Maximum numbers of iterations $\kappa^{\max}, \kappa^{\text{lo,max}}, \kappa^{\text{up,max}}$.
For $a \in \mathcal{A}, t \in \mathcal{T}$, initialize the parameters of the mean function $\mu^{\text{lo}}_{a,t}$ of the
    operating actor and the value function $\widehat{V}^{\text{lo}}_{a,t}$ of the operating critic: $\phi^{\text{lo}}_{a,t} = \mathbf{0}, \omega^{\text{lo}}_{a,t} = \mathbf{0}$.
For $(i,j) \in \mathcal{L}^{\text{tie}}, t \in \mathcal{T}$, initialize the parameters of the mean function $\mu^{\text{up}}_{(i,j),t}$
    of the interconnection actor and the value function $\widehat{V}^{\text{up}}_{(i,j),t}$ of the interconnection
    critic: $\phi^{\text{up}}_{(i,j),t} = \mathbf{0}, \omega^{\text{up}}_{(i,j),t} = \mathbf{0}$.
for $\kappa = 1, 2, \ldots, \kappa^{\max}$ do
    #Operating Agents Training
    for $\kappa^{\text{lo}} = 1, 2, \ldots, \kappa^{\text{lo,max}}$ do
        Initialize the partial states $\widehat{\mathbf{s}}^{\text{lo}}_{a,0}$, $\forall a \in \mathcal{A}$.
        $\mathbf{s}^{\text{up}}_{(i,j),0} = \left[\widehat{\mathbf{s}}^{\text{lo}\top}_{i,0}, \widehat{\mathbf{s}}^{\text{lo}\top}_{j,0}\right]^\top$, $\forall (i,j) \in \mathcal{L}^{\text{tie}}$.
        $T_{(i,j),0} = \mu^{\text{up}}_{(i,j),0}(\mathbf{s}^{\text{up}}_{(i,j),0})$, $\forall (i,j) \in \mathcal{L}^{\text{tie}}$.
        $\mathbf{s}^{\text{lo}}_{a,0} = \left[\widehat{\mathbf{s}}^{\text{lo}\top}_{a,0}, (\mathbf{M}^{\text{tie}}_a \mathbf{T}_0)^\top\right]^\top$, $\forall a \in \mathcal{A}$.
        for $t \in \mathcal{T}$ do
            $\mathbf{u}^{\text{lo}}_{a,t} \sim \pi^{\text{lo}}_{a,t}(\cdot \mid \mathbf{s}^{\text{lo}}_{a,t})$, $\forall a \in \mathcal{A}$.
            $\mathbf{s}^{\text{lo}}_{a,t+1} \sim \mathcal{P}^{\text{lo}}_{a,t}(\cdot \mid \mathbf{s}^{\text{lo}}_{a,t}, \mathbf{u}^{\text{lo}}_{a,t})$, $\forall a \in \mathcal{A}$.
            Perform (15) for area $a$, $\forall a \in \mathcal{A}$.
        end for
        ${\sigma^{\text{lo}}}^2 = \gamma^{\text{lo}} \cdot {\sigma^{\text{lo}}}^2$.
    end for
    # Interconnection Agents Training
    for $\kappa^{\text{up}} = 1, 2, \ldots, \kappa^{\text{up,max}}$ do
        Initialize the partial states $\widehat{\mathbf{s}}^{\text{lo}}_{a,0}$, $\forall a \in \mathcal{A}$.
        $\mathbf{s}^{\text{up}}_{(i,j),0} = [\widehat{\mathbf{s}}^{\text{lo}\top}_{i,0}, \widehat{\mathbf{s}}^{\text{lo}\top}_{j,0}]^\top$, $\forall (i,j) \in \mathcal{L}^{\text{tie}}$.
        for $t \in \mathcal{T}$ do.
            $T_{(i,j),t} \sim \pi^{\text{up}}_{(i,j),t}(\cdot \mid \mathbf{s}^{\text{up}}_{(i,j),t})$, $\forall (i,j) \in \mathcal{L}^{\text{tie}}$.
            $\mathbf{s}^{\text{up}}_{(i,j),t+1} \sim \mathcal{P}^{\text{up}}_{(i,j),t}(\cdot \mid \mathbf{s}^{\text{up}}_{(i,j),t}, T_{(i,j),t})$,
                                                        $\forall (i,j) \in \mathcal{L}^{\text{tie}}$.
            Perform (18) for tieline $(i,j)$, $\forall (i,j) \in \mathcal{L}^{\text{tie}}$.
        end for
        ${\sigma^{\text{up}}}^2 = \gamma^{\text{up}} \cdot {\sigma^{\text{up}}}^2$.
    end for
end for
```

$\mathcal{L}^{\text{tie}} = \{(2,1), (3,1), (3,2)\}$ and $(2,1) \in \mathcal{N}_1 \times \mathcal{N}_2, (3,1) \in \mathcal{N}_1 \times \mathcal{N}_3,$ and $(3,2) \in \mathcal{N}_2 \times \mathcal{N}_3$. All tieline capacity limits are uniformly set to 150 [MW] while for transmission lines $(1,2), (1,3)$ and $(2,3)$, in each area $a \in \mathcal{A}$, the capacity limits are 35 [MW], 100 [MW] and 100 [MW], respectively. Capacity limits and cost/utility function coefficients of generators/elastic loads are presented in Tables III and IV, and operational limits of energy storage devices are presented in Table V, all in the Appendix. The multi-area scheduling is performed through hourly decision-making over a daily horizon, i.e., with $T = 24$ [hr].

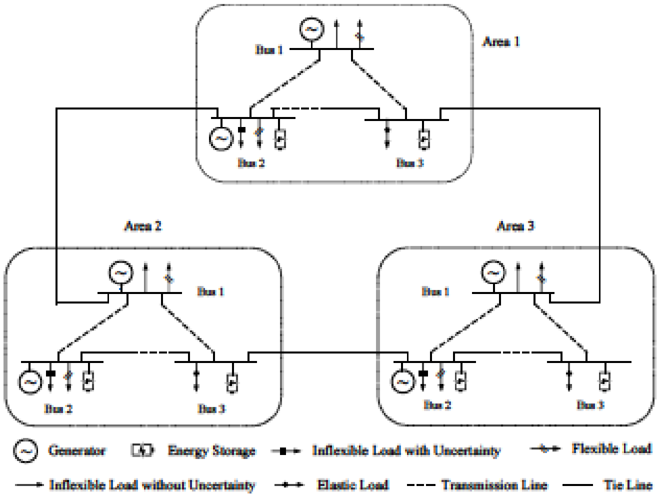

Fig. 4. Three-area test network.

2) Loads and Perturbations: The combined total of flexible and inflexible loads of buses 1 and 2 in each area are provided in Fig. 5, with 80% of the load designated as inflexible and the remaining 20% as flexible. Without loss of generality, we assume that the load at bus 2 in each area is deducted the day-ahead solar power forecast. Thereby, net-load perturbations are introduced to bus 2 of each area $a \in \mathcal{A}$ to model the combined uncertainty originating from load and solar power generation, which adheres to the following distribution:

$$\epsilon_{2,a,t} \sim \begin{cases} \mathcal{N}(0, 10^2), & \text{if } t \leq 9 \text{ or } t \geq 18, \\ \mathcal{N}(0, 5^2) + \epsilon_{2,a,t-1}, & \text{otherwise.} \end{cases} \tag{21}$$

The perturbations shown in Fig. 6 are drawn from $W = 10{,}000$ randomly generated scenarios, where each scenario consists of perturbations for each of the 24 hours. We evaluate the policies derived from Algorithm 1 using these scenarios where for each scenario, indexed by $w = 1, 2, \ldots, W$, we represent the realizations of the perturbations as $\{\epsilon^{(w)}_{a,t}\}_{a \in \mathcal{A}, t \in \mathcal{T}}$.

3) Function Approximation and Parameter Selection: In our experiments, all policies and approximated value functions employ the linear function approximation technique with second-order polynomial features. The Gaussian policies' variances, ${\sigma^{\text{lo}}}^2$ and ${\sigma^{\text{up}}}^2$, are initialized at 0.3 and 0.1, respectively. Both the lower and upper variance reduction factors are set at $\gamma^{\text{lo}} = \gamma^{\text{up}} = 0.999999$. We select a maximum iteration count of $\kappa^{\max} = 30{,}000$ for the outer loop, and the numbers of iterations for inner loops are set to $\kappa^{\text{lo,max}} = \kappa^{\text{up,max}} = 20$. The learning rates for operating agents are selected as $\beta^{\text{lo,ac}} = 0.001$ and $\beta^{\text{lo,cr}} = 0.01$. In contrast, the interconnection agents adopt learning rates of $\beta^{\text{up,ac}} = 0.0001$ and $\beta^{\text{up,cr}} = 0.001$. The hyperparameters were selected using grid search and established practices within the domain of actor-critic algorithms. For example, the learning rates of actors are significantly smaller (10 times smaller) than those of the corresponding critics.

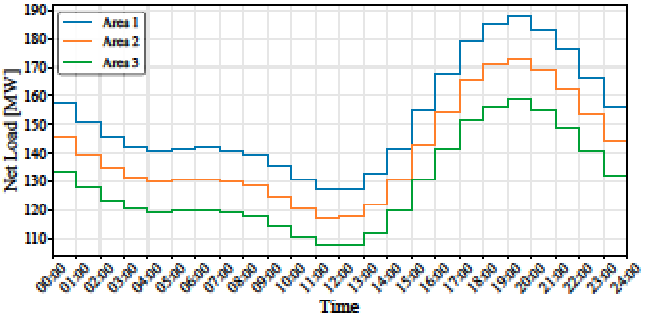

Fig. 5. Combined total of flexible and inflexible loads at buses 1 and 2 for the three areas (buses 1 and 2 in each area share identical loads.).

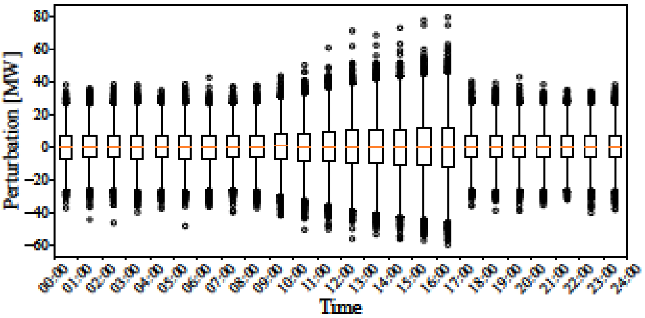

Fig. 6. Box-and-whisker plot for the distribution of perturbations. The line inside the box represents the median value of the perturbations. The bottom and top edges of the box represent the first (25th percentile) and third (75th percentile) quartiles, respectively. The whiskers extend to the minimum and maximum values within 1.5 times the interquartile range. Outliers, represented by circles, are individual perturbations that fall beyond the whiskers.

B. Benchmark Solutions

To provide a basis for comparison, we solve the multi-area power scheduling problem in (14) for each scenario using MPC. By defining $\tau \in \mathcal{T}$ as the start time of the MPC receding horizon, the solution to (14) with a specified perturbation is denoted by $[(\hat{\mathbf{X}}_{a,t}^{(w,\tau)})_{a\in\mathcal{A}}, \hat{\mathbf{T}}_t^{(w,\tau)}]_{t\in\mathcal{T}}^{\top}$ while the perturbation takes the following form:

$$\epsilon_{a,t} = \begin{cases} \epsilon_{a,t}^{w}, & \text{if } t \leq \tau, \\ \mathbb{E}\left[\epsilon_{a,t} \mid \epsilon_{a,\tau} = \epsilon_{a,\tau}^{w}\right], & \text{otherwise}, \end{cases}, \forall a \in \mathcal{A}, t \in \mathcal{T},$$

implying a definite realization for $t \leq \tau$ and uncertain values conditioned on perturbation at τ for $t > \tau$. Additionally, the problem is subject to constraints:

$$\mathbf{X}_{a,t} = \hat{\mathbf{X}}_{a,t}^{(w,\tau-1)}, \ \forall a \in \mathcal{A}, \forall t \leq \tau,$$
$$\mathbf{T}_t = \hat{\mathbf{T}}_t^{(w,\tau-1)}, \ \forall t \leq \tau,$$

to ensure each receding horizon problem is initialized at the optimal solution of the preceding problem. Consequently, we adopt $[(\hat{\mathbf{X}}_{a,t}^{(w,24)})_{a\in\mathcal{A}}, \hat{\mathbf{T}}_t^{(w,24)}]_{t\in\mathcal{T}}^{\top}$ as the solution from MPC.

The choice for this comparison stems from the particular constraints of the setting, where the perturbation $\epsilon_{a,t}$ is accessible only up to time t. It is worth highlighting that the solution from our proposed learning-based algorithm is produced under a more restrictive setting compared to MPC, that is, operating in a decentralized fashion and lacking comprehensive knowledge about the distribution of perturbations. As a consequence, the operating agents do not have access to either the perturbations occurring in other areas or anticipated future values.

TABLE I
AVERAGE NET OPERATION COSTS ± STANDARD DEVIATION [$]

ρ	Algorithm 1	MPC	Centralized RL	Gap
0.95	85,617 ± 145	83,084	84,471 ± 100	3.0% ± 0.2%
1	96,171 ± 333	92,587	94,252 ± 145	3.9% ± 0.4%
1.05	107,357 ± 408	102,330	104,305 ± 136	4.9% ± 0.3%

Additionally, we compare our proposed algorithm with a centralized RL approach. This centralized approach treats the entire system as a unified area addressed by a single operating agent. However, the statistical characteristics of the perturbation remain unknown to the agent.

C. Simulation Results

The average net operation costs over 10,000 scenarios derived from Algorithm 1, as well as MPC and centralized RL solutions, are provided in Table I where, aside from the original load provided in Fig. 5, two additional cases are also examined with nodal loads scaled by factors $\rho = 0.95$ and $\rho = 1.05$ to enable assessing the algorithm's sensitivity to overall loading. The means and standard deviations are calculated from 10 random seeds. For the original load, we observe an average of 3.9% gap in the difference in average net operation costs between the solution obtained from the proposed algorithm and that from MPC. As MPC benefits from full knowledge of the perturbations' distribution and operates in a centralized fashion, we consider the solution from MPC to be the best achievable one and consequently, being 3.9% away from it is a desirable outcome. Additionally, the disparity in average costs between centralized RL and MPC highlights the advantage of having complete knowledge of the perturbation distribution. Similarly, the difference in average costs between MARL and centralized RL underscores the benefits of the centralized approach. In addition, the gap becomes more significant as the load increases ($\rho = 1.05$) possibly due to greater likelihood of a power shortage when only local information is accessible. This conjecture is supported by the observation that the gap in average costs between MPC and centralized RL remains stable, staying within a 2% range. Figure 7 indicates that the average net operation cost converges to a value near that of the solution derived from MPC for the case of $\rho = 1$. Furthermore, it is observed that MARL converges faster than centralized RL.

It is important to emphasize the fact that the computational overhead of the MPC solution renders it an impractical alternative for real-time operation of power-exchange between many areas with multiple nodes. In contrast, once trained, the decentralized RL policies can be readily used in the real-time operation of such settings. In Table II, we present a comparison of computation times between the two methods, i.e., the application of trained RL policies and the solution of

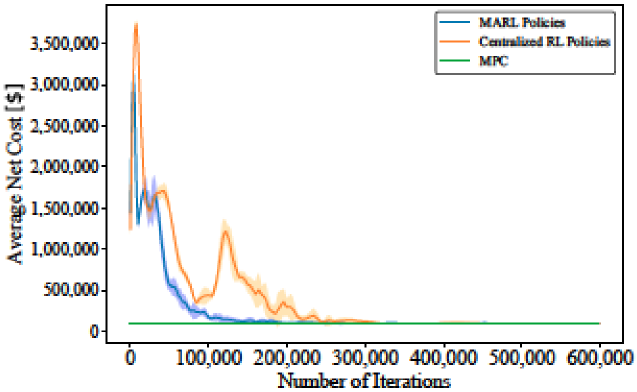

Fig. 7. Convergence of the average net operation cost over training iterations. The x-axis represents the number of iterations for updating the operating agents.

TABLE II
COMPUTATION TIMES [SEC] FOR 10,000 SCENARIOS

Trained MARL Policies	MPC
138.1	3,066.7

the MPC. Both methods were executed on a machine powered by an Apple M2 CPU, for 10,000 individual scenarios, where the MPC model is solved using Gurobi solver. The trained RL policies process approximately 22.2 times faster, taking 138.1 [sec], compared to MPC via Gurobi, which takes 3,066.7 [sec].

VI. CONCLUDING REMARKS

In this paper, we developed a decentralized MARL algorithm to address multi-area power exchange problems. In the decentralized scheme, two types of agents—operating and interconnection agents—cooperate to minimize the overall net operation cost of the system, with only limited information being shared among them. The proposed algorithm adopts a bi-level structure. In the upper-level problems, the interconnection agents determine the flow on tielines. Meanwhile, in the lower-level problems, the operating agents make local decisions for each area. The policies from our proposed algorithm demonstrate excellent performance in the test case, deviating by less than 5% from the centralized MPC solution. Additionally, our algorithm exhibits scalability, attributed to the limited information exchange among agents and the rapid implementation once the policies are trained. In future research, we plan to develop a MARL algorithm where agents predict the responses of their neighbors to actions, facilitating a more autonomous and practical training process.

APPENDIX

A. Generator Data

Generator cost functions are deemed quadratic with c^{q}, c^{l} and c^{o} respectively referring to quadratic, linear, and constant term coefficients. Capacity limits and cost function coefficients for individual generators of all areas are presented in Table III.

TABLE III
PARAMETERS OF GENERATORS

Area	Bus	$\overline{P}$ [MW]	c^{q} [\$/(MWh)^2]	c^{l} [\$/MWh]	c^{o} [\$/h]
1	1	200	0.02	2	20
1	2	250	0.025	1.5	15
2	1	200	0.02	2	20
2	2	250	0.025	1.5	15
3	1	200	0.02	2	20
3	2	250	0.025	1.5	15

B. Elastic Load Data

Elastic load utility functions are deemed quadratic with h^{q}, h^{l}, and h^{o} respectively referring to quadratic, linear, and constant term coefficients. Capacity limits and utility function coefficients for individual generators of all areas are presented in Table IV.

TABLE IV
PARAMETERS OF ELASTIC LOADS

Area	Bus	$\overline{L}^{\mathrm{e}}$ [MW]	h^{q} [\$/(MWh)^2]	h^{l} [\$/MWh]	h^{o} [\$/h]
1	3	250	−0.02	12	5
2	3	250	−0.02	12	5
3	3	250	−0.02	12	5

C. Energy Storage Data

Power and energy capacity limits and charge/discharge efficiency of energy storage devices of all areas are presented in Table V.

TABLE V
PARAMETERS OF ENERGY STORAGE DEVICES

Area	Bus	$\overline{E}$ [MWh]	$\overline{P}^{\mathrm{c}}$ [MW]	$\overline{P}^{\mathrm{d}}$ [MW]	η^{c}	η^{d}
1	2	160	80	80	0.9	0.95
1	3	120	60	60	0.9	0.95
2	2	160	80	80	0.9	0.95
2	3	120	60	60	0.9	0.95
3	2	160	80	80	0.9	0.95
3	3	120	60	60	0.9	0.95

REFERENCES

[1] Y.-Y. Lee and R. Baldick, "A frequency-constrained stochastic economic dispatch model," *IEEE Transactions on power systems*, vol. 28, no. 3, pp. 2301–2312, 2013.

[2] L. Xu, S. Wang, and R. Tang, "Probabilistic load forecasting for buildings considering weather forecasting uncertainty and uncertain peak load," *Applied energy*, vol. 237, pp. 180–195, 2019.

[3] E. Roos and D. den Hertog, "Reducing conservatism in robust optimization," *INFORMS Journal on Computing*, vol. 32, pp. 1109–1127, 2020.

[4] M. Pereira and L. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical Programming*, vol. 52, pp. 359–375, 1991.

[5] G. Lan, "Complexity of stochastic dual dynamic programming," *Mathematical Programming*, pp. 1–38, 2020.

[6] J. Han, L. Yan, and Z. Li, "A task-based day-ahead load forecasting model for stochastic economic dispatch," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5294–5304, 2021.

[7] A. Stratigakos, S. Camal, A. Michiorri, and G. Kariniotakis, "Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy," *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4696–4708, 2022.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[9] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Trans. on Smart Grid*, vol. 9, no. 4, pp. 2574–2594, Jul. 2018.

[10] S. Gupta, R. Hazra, and A. Dukkipati, "Networked multi-agent reinforcement learning with emergent communication," *arXiv preprint arXiv:2004.02780*, 2020.

[11] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5872–5881.

[12] T. Chu, S. Chinchali, and S. Katti, "Multi-agent reinforcement learning for networked system control," *arXiv preprint arXiv:2004.01339*, 2020.

[13] C. Feng and A. L. Liu, "Networked multiagent reinforcement learning for peer-to-peer energy trading," *arXiv preprint arXiv:2401.13947*, 2024.

[14] Z. Yan and Y. Xu, "A multi-agent deep reinforcement learning method for cooperative load frequency control of a multi-area power system," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4599–4608, 2020.

[15] L. Ding, Z. Lin, X. Shi, and G. Yan, "Target-value-competition-based multi-agent deep reinforcement learning algorithm for distributed nonconvex economic dispatch," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 204–217, 2022.

[16] L. Yu, D. Li, and N. Li, "Offline economic dispatch for multi-area power system via hierarchical reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 152, p. 109195, 2023.

[17] M. R. Salehizadeh and S. Soltaniyan, "Application of fuzzy q-learning for electricity market modeling by considering renewable power penetration," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 1172–1181, 2016.

[18] I. Boukas, D. Ernst, and B. Cornélusse, "Real-time bidding strategies from micro-grids using reinforcement learning," 2018.

[19] W. Liu, P. Zhuang, H. Liang, J. Peng, and Z. Huang, "Distributed economic dispatch in microgrids based on cooperative reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2192–2203, 2018.

[20] L. Xi, J. Chen, Y. Huang, Y. Xu, L. Liu, Y. Zhou, and Y. Li, "Smart generation control based on multi-agent reinforcement learning with the idea of the time tunnel," *Energy*, vol. 153, pp. 977–987, 2018.

[21] A. Younesi, H. Shayeghi, and P. Siano, "Assessing the use of reinforcement learning for integrated voltage/frequency control in ac microgrids," *Energies*, vol. 13, no. 5, p. 1250, 2020.

[22] C. Chen, M. Cui, F. Li, S. Yin, and X. Wang, "Model-free emergency frequency control based on reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2336–2346, 2020.

[23] L. Yin, C. Zhang, Y. Wang, F. Gao, J. Yu, and L. Cheng, "Emotional deep learning programming controller for automatic voltage control of power systems," *IEEE Access*, vol. 9, pp. 31 880–31 891, 2021.

[24] J. Yang and C. Su, "Robust optimization of microgrid based on renewable distributed power generation and load demand uncertainty," *Energy*, vol. 223, p. 120043, 2021.

[25] T. P. Chang, "Estimation of wind energy potential using different probability density functions," *Applied Energy*, vol. 88, no. 5, pp. 1848–1856, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261910004733

[26] ——, "Performance comparison of six numerical methods in estimating weibull parameters for wind energy application," *Applied Energy*, vol. 88, no. 1, pp. 272–282, 2011.

[27] A. N. Çelik, A. Makkawi, and T. Muneer, "Critical evaluation of wind speed frequency distribution functions," *Journal of renewable and sustainable energy*, vol. 2, no. 1, 2010.